Apprentissage par Renforcement et Théorie des Jeux pour la coordination de Systèmes Multi-Agents

Alain Dutech, Raghav Aras, François Charpillet

Rubrique

Coordination par les Jeux Stochastiques

Apprentissage par Renforcement et Théorie des Jeux pour la coordination de Systèmes Multi-Agents

Alain Dutech — Raghav Aras — François Charpillet

LORIA - INRIA
615, rue du jardin botanique
Villers les Nancy
France
{dutech,aras,charp}@loria.fr

RÉSUMÉ. Nous présentons les principaux algorithmes d'apprentissage par renforcement visant à coordonner des systèmes multi-agents en s'appuiant sur des outils et des notions tirés de la Théorie des Jeux. Les limitations des ces approches sont évoquées et discutées afin d'essayer de dégager des problématiques prometteuse pour ce champs de recherche. Nous y discutons en particulier la pertinence des équilibres de Nash et des jeux à information partielle.

ABSTRACT. This article presents the main reinforcement learning algorithms that aim at coordinating multi-agent systems by using tools and formalisms borrowed from Game Theory. Limits of these approaches are studied and discussed in order to draw some promising lines of research for that particular field. We argue more deeply around the central notions of Nash equilibrium and games with imperfect monitoring.

MOTS-CLÉS : Systèmes Multi-Agents, Coordination, Théorie des Jeux, Apprentissage par Renforcement

KEYWORDS : Multi-Agent Systems, Coordination, Game Theory, Reinforcement Learning

1. Introduction

Les méthodes d'apprentissage par renforcement (AR) [16] et leur formalisation par les processus décisionnels de Markov (MDP[1]) [15] ont pris une place importante dans les travaux de recherche en intelligence artificielle. Plus récemment, leur utilisation a été élargie aux systèmes multi-agents (SMA), en tant que méthode de construction automatique mais aussi en tant que formalisme. Cet élargissement a conduit assez naturellement à utiliser et à s'inspirer du bagage mathématique de la Théorie des Jeux (TdJ) [14] pour étendre les outils et formalismes classiques de l'AR.

Nous allons plus précisément nous attacher au problème de la coordination et de la coopération rationnelle d'agents. La question posée est alors de savoir si les algorithmes qui s'appuient sur la combinaison des formalismes des MDP et de la TdJ permettent effectivement à des agents de se coordonner, voire de coopérer.

Le but de cet article est en fait triple. Premièrement, nous voulons présenter de manière synthétique les travaux majeurs de ce courant de recherche qui allie AR et TdJ. Cette présentation essaiera aussi de dégager les principaux concepts mis en oeuvre en apprentissage par renforcement. Deuxièmement, nous voulons essayer de mettre en avant les problèmes et les limitations des travaux actuels. De cette discussion, nous ferons émerger les problématiques qui nous semblent les plus intéressantes et qui sont, ou seront, les points de focalisations des recherches futures. Nous discuterons notamment de la pertinence des équilibres de Nash et de l'intérêt de pouvoir traiter des problèmes d'information partielle. Enfin, nous proposerons des pistes, plus ou moins défrichées, pour avancer dans les directions qui nous paraissent les plus prometteuses.

Nous avons donc organisé cet article comme suit. La Section 2 présente le premier formalisme, au sens chronologique, alliant MDP et SMA. Outre la présentation des principes des MDP, cette partie explique comment le problème de coordination devient un problème lié à la TdJ. La Section 3 présente alors les principaux travaux s'appuyant à la fois sur la TdJ et les MDP pour résoudre ce problème de coordination, alors que la Section 4 s'y intéresse quant les agents n'ont que des informations partielles sur leur environnement. Notre point de vue sur les limitations et les problèmes les plus intéressants qui restent à résoudre est explicité en Section 5. Nous y ajoutons des propositions de direction de recherche plus ou moins concrètes en Sections 5.3 et 5.4 avant de conclure.

2. Rationalité multi-agents et coordination

Les travaux de Boutilier [2, 6] sur les processus décisionnels Markoviens *multi-agents* (MMDP) ont été parmi les premiers à poser le problème de la conception de systèmes multi-agents en terme de théorie de la décision.

Un MMDP est un tuple $\langle \mathcal{S}, N, \mathcal{A} = \{\mathcal{A}_i\}_{i \in N}, p, r \rangle$ où : $\mathcal{S}$ et N sont des ensembles finis d'états et d'agents ; $\mathcal{A}_i$ est un ensemble fini d'action pour l'agent i ; $p : \mathcal{S} \times \mathcal{A}_1 ... \times \mathcal{A}_n \times \mathcal{S} \rightarrow [0, 1]$ est une fonction de transitions stochastique entre les états et $r : \mathcal{S} \rightarrow \mathbb{R}$ est une fonction de récompense globale.

Dans ce modèle, le système étant dans un état s_t au temps t, les agents choisissent une action a_i, ce qui forme une action jointe $a = \{a_i\}$ et qui modifie l'état du système de manière stochastique : $\Pr(s_{t+1}|s_t, a) = p(s_t, a, s_{t+1})$. Les agents, reçoivent alors une

1. pour cet acronyme, comme pour d'autre par la suite, nous utiliserons la version anglaise qui est souvent plus parlante et plus connue

récompense $r_{t+1} = r(s_t, a)$. Le problème posé est de trouver les actions qui maximisent un critère fonction de la récompense reçue, classiquement $E[\sum_{t=0}^{\infty} \gamma^t r_t]$ où $\gamma \in [0, 1[$.

De manière analogue au cadre mono-agent largement étudié (voir [15, 16]), on formalise les décisions des agents sous la forme de politiques d'action $\pi_i : \mathcal{S} \rightarrow \Delta(\mathcal{A}_i)$ ($\Delta(.)$ dénotant une distribution). Pour chaque politique d'action jointe $\pi = \{\pi_i\}$, on peut associer une *fonction valeur* $Q(s, a)$ qui, pour chaque état, est l'espérance de sa valeur (au sens du critère précédent) si les agents choisissent d'abord l'action a avant de suivre la politique π. Dans ce cadre, on sait qu'il existe des politiques jointes optimales π^* dont la fonction valeur optimale Q^* vérifie l'équation de Bellman :

$$Q^*(s, a) = r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s, a, s') \max_{a' \in \mathcal{A}} Q^*(s', a') \quad [1]$$

On a alors que $\pi^*(s) = \text{argmax}_{a \in \mathcal{A}} Q^*(s, a)$.

Chaque agent peut individuellement, en utilisant des algorithmes classiques [15, 16], trouver des politiques optimales. Néanmoins, s'il existe plusieurs politiques optimales, les agents *doivent* choisir, de manière indépendante, la même politique optimale : ils doivent se coordonner. Boutilier montre qu'apprendre à se coordonner revient à apprendre des équilibres dans des jeux à n joueurs.

3. MDP multi-agents et recherche d'équilibres

3.1. Q-learning Multi-agent générique

La forme générale des algorithmes d'apprentissage par renforcement pour apprendre des équilibres dans des jeux est la suivante : (1) les agents, dans un état s, choisissent leurs actions $a_1, ..., a_n$; (2) l'état du système est modifié en s' et les agents reçoivent une récompense $r_1, ..., r_n$; (3) les équations 2 et 3 permettent aux agents de ré-estimer leur politique optimale ($\alpha \in]0, 1[$ est un coefficent d'apprentissage) ; (4) on revient au point (1) avec $s = s'$

$$V_i(s') \quad \longleftarrow \quad f_i(Q_1(s', .), ..., Q_n(s', .)) \quad [2]$$

$$Q_i(s, a) \quad \longleftarrow \quad (1 - \alpha) Q_i(s, a) + \alpha[r_i + \gamma V_i(s')] \quad [3]$$

Les différents travaux que nous allons passer en revue se différencient principalement par la forme générale que peuvent prendre les fonctions récompenses et surtout par les fonctions f_i qui choisissent, à partir des informations courantes sur les matrices de récompenses de *tous les joueurs*, les équilibres à utiliser. On peut alors remarquer que, de manière générale, les agents *doivent* pouvoir observer les actions et les récompenses des autres agents pour maintenir cette connaissance sur les autres.

3.2. Déclinaisons de l'algorithme générique

Avec le *minimax-Q* [11], Littman s'est d'abord intéressé à des jeux à somme nulle à deux joueurs. Les équilibres y sont les politiques où chaque joueur cherche à maximiser ses gains en minimisant celui de l'autre. La fonction f_i est alors :

$$V_1(s) = \max_{\sigma_1 \in \Delta(\mathcal{A}_1)} \min_{a_2 \in \mathcal{A}_2} Q_1(s, (\sigma_1, a_2)) = -V_2(s)$$

Néanmoins, en dépit de ses garanties de convergence vers des politiques optimales (avec des limitations forte sur la nature du jeux, voir [13]), le cadre applicatif de cet algorithme

le rend impropre à la recherche de coordination car, à cause de la forme des récompenses, les agents ont ici des but totalement opposés.

Si l'on s'intéresse au cadre plus complet des jeux à somme générale (les récompenses individuelles peuvent y être quelconque), une généralisation des travaux précédents à été proposée par Hu et Wellman [9] avec l'algorithme *Nash-Q*. Le principe est de s'appuyer sur les équilibres de Nash [14] pour coordonner les agents. La fonction de choix d'équilibre devient alors $V_i(s) \in \mathrm{NASH}_i(Q_1(s,.), ..., Q_n(s,.))$ où NASH_i est l'ensemble des équilibres de Nash du jeux dans l'état s. Il apparaît rapidement que les jeux possédant plusieurs équilibres différents peuvent poser problème. De fait, comme établit dans [10], la convergence de cet algorithme n'est garantie que pour des jeux possédant des caractéristiques très particulières.

Pour s'affranchir un peu de la nécessité pour chaque agent de pouvoir observer les actions et les récompenses des autres afin de trouver les équilibres de Nash, Littman [12] a proposé une version plus spécialisée du *Nash-Q* appelée *Friend-or-Foe Q-Learning* (FoFQ). Si les joueurs doivent savoir à quelle forme de jeu ils sont confrontés, ils n'ont alors plus besoin de tout savoir sur les autres joueurs.

Dans l'algorithme *Correlated Q-Learning* (CorrQ) proposé par Greenwald et Hall [7] la fonction f_i de sélection d'équilibre ne cherche plus des équilibres de Nash (ou approchés) mais cherche des *équilibres corrélés* qui sont plus adéquats quand on veut faire coopérer des agents. Ce sont des équilibres sur les actions jointes (ce qui suppose une certaine confiance entre les agents) mais qui permettent de recevoir une récompense au moins égale au meilleur équilibre de Nash.

De manière pratique, la recherche d'équilibre μ (où $\mu(\pi)$ est la probabilité qu'une politique *déterministe* soit recommandée) se fait par programmation linéaire avec le système d'inégalité suivant :

$$\sum_{\pi_{-i} \in \Pi_{-i}} \mu(\pi)(r_i(\pi) - r_i(\pi_{-i}, \pi_i)) \quad \geq 0 \quad \forall i \in N, \forall \pi \in \Pi_i, \forall \pi_i \in \Pi_i \qquad [4]$$

Outre le fait qu'il peut exister plusieurs équilibre corrélés (et qu'il faut donc des heuristiques pour choisir parmi ces équilibres), l'algorithme nécessite aussi que chaque agent puisse observer les actions et les récompenses des autres agents.

3.3. Algorithme "Win or Learn Fast"

L'algorithme *WoLF* (pour "Win or Learn Fast") de Bowling et Veloso [3] sort du cadre générique de la Section 3.1. Cet algorithme s'inspire des algorithmes d'itération de la politique. Pour une politique *stochastique* donnée π_i, un joueur peut calculer une estimation de la valeur de cette politique. Après chaque interaction avec le jeu, le joueur peut utiliser cette estimation de la valeur pour améliorer sa politique actuelle en augmentant d'une quantité δ la probabilité que l'action "optimale" soit choisie à l'avenir. Le coefficient d'apprentissage δ dépend du fait que le joueur soit en train de gagner (δ sera alors faible) ou de perdre (δ plus grand). Plus précisément :

$$\delta \quad = \quad \begin{cases} \delta_w & \text{si } \sum_{a'_i} \pi_i(s, a'_i) Q(s, a'_i) > \sum_{a'_i} \bar{\pi}_i(s, a'_i) Q(s, a'_i) \\ \delta_l & \text{sinon} \end{cases} \qquad [5]$$

où $\delta_w < \delta_l$ et $\bar{\pi}_i$ est la politique moyenne suivie par l'agent i.

Dans le cas précis d'un jeu itéré à deux joueurs avec deux actions, si les deux joueurs utilisent l'algorithme *WoLF*, alors leurs politiques vont converger vers un équilibre de

Nash. La version théorique de l'algorithme nécessite que chaque agent connaisse sa fonction de récompense, la politique de l'autre joueur. En pratique, cet algorithme a été validé sur des jeux à deux joueurs plus complexes (plus d'actions, jeux stochastiques) où il a convergé vers des politiques qui étaient les meilleurs réponses possibles aux stratégies adverses.

4. Jeux avec information partielle

Le cas des jeux avec informations partielle, où les joueurs ne connaissent pas exactement l'état ni les actions et récompenses reçue par les autres joueurs, est notablement plus complexe. Du point de vue de l'apprentissage par renforcement, ces jeux se rapprochent des processus décisionnels Markoviens partiellement observables (POMDP, voir [4]).

4.1. Formalisme

Les jeux stochastiques partiellement observables (POSG) sont définis par $\langle \mathcal{S}, N, \mathcal{A} = \{\mathcal{A}_i\}, \{\Omega_i\}, p, r \rangle$. On a donc ajouté, pour chaque agent, un ensemble Ω_i d'observations et les fonctions de transitions p sont modifiées pour fournir, en plus de la probabilité de l'état suivant, la probabilité des observations :$p(s_t, a, s_{t+1}, o = (o_i)) = \Pr(s_{t+1}, o | s_t, a)$. Quant aux joueurs, ils ne connaissent pas l'état mais peuvent seulement l'observer.

4.2. Elagage itératif

Il est intéressant de noter que les travaux les plus avancés sur le sujet, ceux de Hansen, Bernstein et Zilberstein [8] ont vraiment rapproché le domaines de l'apprentissage par renforcement (plus particulièrement des POMDP) et de la théorie des jeux.

L'idée principale pour résoudre un POMDP est de se ramener à un cas totalement observable en ne considérant plus les états du processus mais des distributions de probabilités sur ces états. On parle alors d'état de croyance $b(s) = \Pr(s_t = s)$ qui sont définis sur $\Delta(\mathcal{S})$. La valeur d'une politique π est une fonction linéaire par morceau qui peut s'exprimer comme $V(b) = max_j \sum_{s \in \mathcal{S}} b(s) v_j(s)$ où $\mathcal{V} = \{v_1, ..., v_k\}$ est un ensemble de vecteurs. Ces vecteurs sont tous des combinaisons des vecteurs représentatifs des sous-politiques de π. En partant de politiques simples (une seule action), un calcul récursif permet alors de calculer la politique optimale. A chaque étape, un élagage des vecteurs inutiles permet d'éviter une explosion combinatoire du nombre de vecteurs [5].

Cette idée a été étendue aux POSG par Hansen et al. [8] (*DP-POSG*), mais avec une notion de croyance adaptée au cadre multi-agent. Ainsi, pour un joueur i, un état de croyance b_i est définit comme étant une distribution sur $\mathcal{S} \times \Pi_{-i}$ où Π_{-i} est l'ensemble des politiques possibles des autres agents. Ainsi, en une étape similaire au test de dominance utilisé dans les POMDP, en s'appuyant aussi sur la programmation linéaire, il est possible de mettre en oeuvre une étape d'élagage dans les POSG qui élimine les politiques déterministes π_i qui sont très faiblement dominées, c'est-à-dire où il existe une politique stochastique μ_i telle que :

$$V_i(s, (\mu_i, \pi_{-i})) \quad \geq \quad V_i(s, (\pi_i, \pi_{-i})),\ \forall s \in S,\ \forall \pi_{-i} \in \Pi_{-i} \qquad [6]$$

Il est alors prouvé qu'un tel algorithme directement inspiré de la programmation dynamique, met en oeuvre une méthode d'élimination itérative des politiques très faiblement dominées, méthode bien connue de la théorie des jeux [14]. On génère ainsi un ensemble de politique qui *peut* contenir tous les équilibres de Nash.

Algo	Connaît $\mathcal{S}$	Connaît $\mathcal{A}+r$	r restreint	Equ. Nash	Garanties Théoriques
minimaxQ	.	.	.	.	+
NashQ	.	.	+	.	$\sim$
FoFQ	.	+	.	.	+
WoLF	.	+	$\sim$	.	$\sim$
CorrQ	.	.	+	+	.
DP-POSG	+	+	+	.	$\sim$
BCE-Q	+	+	$\sim$	+	.

Tableau 1. *Comparaison des algorithmes évoqués (voir texte). Un '+' signifie que l'algorithme n'est pas dépendant de cette contrainte, un '$\sim$' que cette dépendance est forte mais pas absolue, et un '.' que l'algorithme dépend de cette contrainte.*

5. Discussion et propositions

La Table 5 résume les caractéristiques des algorithmes auxquelles nous allons nous attacher lors de cette discussion.

5.1. Agents omniscients

Les algorithmes qui s'appuient sur les MDP (voir Section 3) sont quasiment tous dépendants du fait que les agents doivent connaître l'état du jeu. Ils doivent aussi pouvoir observer les actions et les récompenses des autres joueurs. Ces contraintes sont difficilement compatibles avec le formalisme des systèmes multi-agents qui s'appuient sur la localité des agents, ce qui implique une connaissance partielle du système et des autres.

L'algorithme DP-POSG de Hansen et Bernstein est une réponse possible à ce problème, mais limitée par sa complexité et sa recherche d'équilibres de Nash, d'ailleurs peu efficaces(voir Section 5.3).

5.2. Coordination, Coopération et Nash

En fait, s'il apparaît évident que le fait de pouvoir traiter des jeux où les agents ont tous la même récompense est nécessaire pour construire des agents coopérants, ce n'est pas suffisant. En effet, on ne peut écarter le cas d'agents légèrement différents voulant tout de même coopérer localement et temporairement, et dans ce cas il faut pouvoir gérer des jeux où chaque agent a une fonction de récompense propre et indépendante des autres agents.

De plus, les équilibres de Nash peuvent être globalement *inefficaces* : il existe de nombreux jeux où l'équilibre de Nash est moins efficace que d'autres choix mais qui nécessitent soit des médiateurs, soit de la communication, soit des agents se faisant confiance pour être choisis par les agents. C'est justement ce que nous attendons d'agent devant coopérer. L'algorithme *Correlated-Q* répond en partie à ce problème, mais nécessite des agents omniscients.

5.3. Equilibres séquentiels dans les POSG

Dans l'algorithme *DP-POSG*, le fait d'éliminer des sous-politiques même très faiblement dominées, fait courir le risque d'éliminer des équilibres de Nash qui sont, eux, globaux. Ce n'est pas gênant si ces équilibres sont en fait irréalistes (voir [14]) mais rien

ne prouve que cela soit le cas. En fait, l'algorithme semblerait mieux adapté à la recherche d'*équilibres séquentiels* qui sont, aussi, des équilibres de sous-politiques.

Nous sommes donc en train de réfléchir à une adaptation de cet algorithme, notamment au travers d'heuristique pour déterminer quels sont les sous-jeux d'un jeu donné (notion rendue complexe par l'observabilité partielle de l'état), car il est alors possible d'utiliser des tests de dominance locaux dans ses sous-jeux sans pour autant éliminer d'équilibres séquentiels. Ce sont des travaux de recherche en cours.

5.4. Communication limitée

Pour une utilisation plus réaliste des jeux comme cadre formel de la coordination multi-agents, les algorithmes proposés ne doivent pas dépendre du fait que les agents puisse observer les actions et les récompenses des autres agents. La recherche de politiques s'appuyant sur la notion d'équilibre de Nash n'est pas non plus satisfaisante pour l'apprentissage de la coordination car les agents se limitent alors seulement à limiter leurs pertes, sans chercher à coopérer.

Nous avons donc proposé un formalisme d'apprentissage appelé *Best Compromise Equilibrium Q-Learning* (BCE-Q) [1] qui respecte ces contraintes dans le cadre général des POSG. Ce formalisme s'appuie sur une communication la plus limitée possible (pour ne pas alourdir plus que de raison le processus d'apprentissage) entre les agents pour mettre en oeuvre un recherche d'équilibre s'inspirant de méthodes des transferts de récompenses entre les agents, méthodes parfois mises en oeuvre dans les jeux de négociation.

Le principe général de cet algorithme est de modifier les récompenses reçues par les agents en fonctions des messages binaires (oui/non) envoyés par les agents et de la récompense données par l'environnement. La forme générale de cette récompense "virtuelle" est $\hat{r}_i = (x|m_{i.s}| + y|m_{i.r}| + z)r_i$ où x, y et z sont des paramètres (à choisir ou apprendre) et $|m_{i.s}$ (resp. $|m_{i.r}|$) est le nombre de messages envoyés (resp. reçus) par l'agent i. Cet algorithme a été testé expérimentalement avec des paramètres fixés manuellement pour des jeux stochastiques et permet de trouver des politiques plus performantes que celles trouvées par *Nash-Q*.

6. Conclusion

Nous avons essayé de faire un tour d'horizon à la fois complet et synthétique des recherches actuelles qui visent à allier la Théorie des Jeux et les Processus Décisionnels de Markov pour coordonner rationnellement des agents. Nombre de ces travaux se sont concentrés sur la possibilité d'apprendre des équilibres de Nash. Nous avons argumenté que, d'une part, ces algorithmes nécessitent des agents omniscients et donc peu réalistes ; d'autre part, les équilibres de Nash ne sont pas forcément adaptés au problèmes de coordination et coopération.

DP-POSG [8] est un des rares algorithmes qui s'attaque au problèmes d'agents ayant des informations limitées. Nous avons alors montré que cet algorithme souffrait de quelques défauts et suggérés des pistes de travail possible pour les corriger. Nous avons aussi proposé un algorithme [1] qui essaie de répondre aux deux critiques formulées plus haut en permettant aux agents une communication limitée et un transfert de récompense.

Ces travaux sont toujours en développement et montrent que les questions de communication et d'information partielles restent des voies de recherche importantes et actuelles.

7. Bibliographie

[1] ARAS, R., DUTECH, A., AND CHARPILLET, F. Cooperation in stochastic games through communication. In *Proc. of the fourth Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS'05)* (Utrech, Nederlands, 2005).

[2] BOUTILIER, C. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge (TARK '96), De Zeeuwse Stromen, The Netherlands* (1996).

[3] BOWLING, M., AND VELOSO, M. Multiagent learning using a variable learning rate. *Artificial Intelligence 136* (2002), 215–250.

[4] CASSANDRA, A. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Brown University, Department of Computer Science, Providence, RI, 1998.

[5] CASSANDRA, A., LITTMAN, M., AND ZHANG, N. Incremental pruning : A simple, fast, exact method for partially observable markov decision processes. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)* (1997).

[6] CLAUS, C., AND BOUTILIER, C. The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI* (1998), pp. 746–752.

[7] GREENWALD, A., AND HALL, K. Correlated Q-learning. In *Proc. of the 20th Int. Conf. on Machine Learning (ICML)* (2003).

[8] HANSEN, E., BERNSTEIN, D., AND ZILBERSTEIN, S. Dynamic programming for partially observable stochastic games. In *Proc. of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)* (2004).

[9] HU, J., AND WELLMAN, M. Multiagent reinforcement learning : theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning, ICML-98* (1998), pp. 242–250.

[10] HU, J., AND WELLMAN, M. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research* (2003).

[11] LITTMAN, M. Markov games as a framework for multi-agent reinforcement learning. In *Proceddings of the eleventh Int. Conference on Machine Learning, San Fransisco, CA* (1994).

[12] LITTMAN, M. Friend-or-foe Q-learning in general-sum games. In *Proc. of the 18th Int. Conf. on Machine Learning (ICML)* (2001).

[13] LITTMAN, M., AND SZEPESVÁRI, C. A generalized reinforcement-learning model : Convergence and applications. In *Proc. of the Thirteenth Int. Conf. on Machine Learning (ICML'96)* (1996).

[14] MYERSON, R. *Game Theory : Analysis of Conflict*. Harvard University Press, 1991.

[15] PUTERMAN, M. *Markov Decision Processes : discrete stochastic dynamic programming*. John Wiley & Sons, Inc. New York, NY, 1994.

[16] SUTTON, R., AND BARTO, G. *Reinforcement Learning*. Bradford Book, MIT Press, Cambridge, MA, 1998.